SYSTEM AND METHOD OF MINING WORLD WIDE WEB CONTENT

RELATED APPLICATIONS

The present application claims the benefit of U.S. Provisional Patent Application No. 60/211,161 entitled "TARGETED WEB MINING METHOD," filed June 13, 2000.

TECHNICAL FIELD OF THE INVENTION

This invention is related in general to the field of Internet appliances and devices. More particularly, the invention is related to a system and method of mining world wide web content.

BACKGROUND OF THE INVENTION

In a fully automated environment, appliances that change the various parameters of the environment can be inked to a control area network (CAN) and a computer-based controller. The appliances may include heating, ventilation and air conditioning (HVAC) systems, lighting systems, audio-visual systems, telecommunications systems, security systems, surveillance systems, and fire protection systems, for example. One or more easy-to-use user interface, such as a touch panel, may be electronically linked to the control area network to accept user input and display current system status. Panja, Inc. of Dallas, Texas designs and manufactures such networked appliance control systems.

These control area networks may further be coupled to the Internet to enable remote access and control. In addition, such connectivity also enables data content available on the World Wide Web (WWW) to become accessible to the users of the control area network. However, because these Internet appliances are typically non-personal computers with substantially different display sizes and formats and the dynamic nature of web pages, the function of downloading and displaying web content is not an easy or straightforward problem to resolve.

SUMMARY OF THE INVENTION

In accordance with the present invention, a system and method of mining web page content for display on an internet appliance are provided which eliminate or substantially reduce the disadvantages associated with prior systems.

In one aspect of the invention, a method of extracting content from a web page includes the steps of loading a template of the web page and parsing at least one set of tags in the template, storing the at least one set of tags in a first hierarchical data structure, and parsing an instance of the web page and extracting content according to the stored tags in the hierarchical data structure. The extracted content is then stored into a second hierarchical data structure, and the extracted content is processed for output.

In another aspect of the invention, a method of extracting content from a targeted web page includes the steps of creating a template of the targeted web page, the template having the at least one set of tags to demarcate web content to be extracted and control content extraction. The at least one set of tags includes tags for flow control, tags for setting a file pointer used during web page parsing, and tags for indicating how demarcated web page content is to be extracted. The method includes parsing the at least one set of tags in the template, storing the parsed at least one set of tags in a first data structure, and then parsing an instance of the web page and extracting content according to the stored tags in the first data structure. The extracted content is stored into a second data structure. The extracted content stored in the second data structure is then accessed and processed for output.

In yet another aspect of the present in invention, a system and method of mining targeted world wide web content are provided. The system includes a template created for each targeted web page, where each template including at least one set of tags for flow control and content extraction. A template parsing engine is operable to parse the at least one set of tags in the template and storing the tags in a first data structure. A page parsing engine is operable to parse an instance of the targeted web page corresponding to the template according to the stored tags in the first data structure, and extracting its content. A second data structure is used to store

the extracted content of the targeted web page. A presentation application is operable to access, process and output the stored content.

BRIEF DESCRIPTION OF THE DRAWINGS

For a better understanding of the present invention, reference may be made to the accompanying drawings, in which:

FIGURE 1 is a simplified top-level block diagram of a system and method of coupling one or more control systems to the Internet constructed according to an embodiment of the present invention;

FIGURE 2 is a more detailed block diagram of a system and method of coupling one or more control systems to the Internet constructed according to an embodiment of the present invention;

FIGURE 3 is a simplified block diagram of the system and method of mining web content according to an embodiment of the present invention;

FIGURE 4 is a more detailed block diagram of the system and method of mining web content according to an embodiment of the present invention;

FIGURE 5 is an exemplary block diagram of a mined content tree structure according to an embodiment of the present invention; and

FIGURE 6 is a block diagram of exemplary objects used during the web content mining operation according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

FIGURE 1 is a simplified top-level block diagram of a system and method 10 of Internet control system which couple one or more control systems to the Internet constructed according to the teachings of the present invention. The implications of employing system and method 10 of the present invention are the ability to communicate with, control, and be controlled by one or more Internet nodes or Internet applications that act as one or more devices in a control system connected by a control area network (CAN). These Internet applications may include web browsers, web server applications of information content providers, and email applications. In other words, the geographical and communication protocol boundaries are transparent between a local control area network and the Internet, so that the Internet, web information content providers and web browser applications become devices in the control system. By definition, a device in the control system can send control commands to and/or receive control messages from a master controller on the control area network. Hereinafter, the word Internet may be also used to refer to an Intranet or the World Wide Web and vice versa.

System 10 includes a control network portal 12 coupled between the Internet 22 and one or more control area networks 30 and 31. Control area networks 30 and 31 are local area networks operating under transport protocols such as Ethernet, and AXLink and PhastLink of Panja Inc. (Dallas, Texas) that interconnect a variety of devices, appliances and/or equipment. The underlying network connectivity 34 may be wired, wireless, power line carriers, or any suitable transmission medium. Coupled to control area networks 30 and 31 are a plurality of devices, appliances and/or equipment, including control area network user interfaces (CAN UI/F) 35, master controllers 36, and Internet appliances 37-39. Some devices may be coupled to control area networks 30 and 31 via additional intermediate communications devices, such as an RS 232 controller (not shown).

Control area network user interface device 35 is any device that is capable of receiving user input and displaying or indicating control network status. For example, a touch panel, a computer terminal with a monitor, keyboard and pointing device, and any device with similar functionalities may serve as control area network user interface 35. As described in detail below, with the use of control area network portal 12 of the present invention, Internet applications are also capable of functioning as control area network user interface devices without the use of custom and dedicated applications on the user's end.

Master controller 36 is generally a CPU-based controller that controls the communications among user interface 35 and Internet appliances 37-39. It is operable to receive user inputs received by user interface devices, such as commands, and instruct the appropriate Internet appliance to act according to the command. Master controller 36 may also poll each device in control area network 30 periodically to monitor its status. The system status and/or the status of each device may be sent to control area network user interface devices for display.

Internet appliances 37-39 are devices that can receive commands from master controller 36 and operate or act according to the command. Internet appliances 37-39 may include equipment that affect or monitor the various parameters of the premises. For example, Internet appliances 37-39 may include heating and air conditioning, lighting, video equipment, audio equipment, sprinklers, security cameras, infrared sensors, smoke detectors, etc. in a residential or commercial control area network. Household appliances, such as a hot tub, fireplace, microwave oven, coffee maker, etc. may also be Internet appliances coupled to the network. Internet appliances 37-39 may also be capable of providing a current status of its operational state to master controller 36, such as on/off, temperature settings, current ambient temperature, light intensity settings, volume settings, threshold settings, and predetermined alphanumeric strings reflective of operational states.

Master controller 36 is also operable to receive user input from nodes of the Internet 22 via control network portal 12. Connected to Internet 22 are content providers 25 and 26, which may also function as control area network user interface devices. Content providers 25 and 26 are typically web servers that generate and provide static and/or dynamic information and content in the form of web pages. Content provider applications executing on the web server are able to access and download data stored in databases (not shown). The web pages are typically developed with hypertext markup language (HTML), and various other scripting languages and programming environments such as Microsoft® Active Server Pages (ASP), Common Gateway Interface (CGI), Internet Server Application Programming Interface (ISAPI), JAVA, ActiveX, Cold Fusion, etc. that make the web pages more dynamic and interactive. Web content is typically formatted and sized for display on computer monitors.

Also connected to the Internet 22 are web browsers 23 and 24 that may also serve as control area network user interfaces. Web browsers 23 and 24 are application programs that can be used to request and download web pages from content providers 25 and 25 and decode the web pages. Web browser applications include NETSCAPE

NAVIGATOR® and MICROSOFT INTERNET EXPLORER®, for example. However, typical web browser applications are operable to display web content on a computer monitor only. Typically, a user executes a web browser application on her personal computer and accesses the World Wide Web via a dial-up connection to an Internet service provider. The Internet or World Wide Web may also be accessed via other means such as cable modems and digital subscriber lines (DSL). The user makes a request for a particular web page or particular web site by entering or specifying a uniform resource locator (URL). The URL is associated with an Internet protocol (IP) address of the specified web site. Every computer connected to the World Wide Web and Internet has a unique IP address. This address is used to route message packets to specific computers and users. Internet protocol or IP is the message transport and communications protocol of the Internet and World Wide Web.

When the web browser requests a certain URL, a connection is first established with a web server of a content provider that is addressed by the URL. A hypertext transport protocol (HTTP) request is then issued to the web server to download an HTML file. The web server receives the request and sends a web page file to the web browser, which decodes the file to display information in specified format on the screen. Web pages with dynamic content provided by gateway interfaces such as CGI and ISAPI are executable applications that are ran by the web server upon user request. The executing gateway application is able to read parameter information associated with the request and generate an output in the form of an HTML file in response to the parameter values. Another way to add dynamic and interactive content to web pages uses ASP. ASP scripts are server-side executable scripts that are directly incorporated in the HTML web pages. Upon request for the page, the web server executes the ASP script in response to input parameter values and generates the web page with dynamic content.

Using control network portal 12, users may access control area networks 30 and 31 via web browsers 23 and 24 accessing web pages provided by control network portal 12 or value-added web pages provided by content providers 25 and 26. For example, a user who has a control area network deployed in her luxury residence to control various aspects of the home environment may use a web browser application to remotely monitor her home. She may change the temperature setting to decrease energy use, for example, because she will be leaving on a business trip straight from work. She may also use the surveillance cameras to visually ensure security has not been breached. She may even be able to remotely program her VCR to record certain favorite programs that will be broadcast while she is away.

An example of value-added web pages provided by content providers is the provision of an interactive version of the television programming web page, www.tvguide.com. A user may request this web page, determine available program choices, and click on a certain program. Options may be provided to enable the user to turn on the television and tune to a particular channel scheduled to broadcast the selected program or to program the VCR to record the selected program.

Another example of value-added web pages provided by content providers is the provision of a secured web page that an electric company may access to slightly raise the temperature settings of the air conditioning systems of its participating customers in anticipation of high demand brown out conditions. Yet another example is a web page that a security company may use to access, monitor and control the security, surveillance and fire protection systems of its customers.

FIGURE 2 is a more detailed block diagram of a system and method 10 of coupling one or more control system to the Internet constructed according to an embodiment of the present invention. Control area network portal 12 may include a web server 13 coupled to the Internet 22. Web server 13 is also coupled to an Internet appliance (IA) server 14, which may also be coupled to a control network server 40. Control network server 40 is coupled to control area network 30 that links several appliances and systems, such as fire protection systems 50, heating, ventilation and air conditioning (HVAC) systems 51, lighting systems 52, audio and visual systems 53, and security systems 54. Control area network 30 is also coupled to user interface devices 55 and master controller 36.

It may be noted that control network portal 12 may be implemented by a single stand-alone system that has sufficient memory and processing power or several separate systems with distinct functions as shown in FIGURE 2. Web server 13 is operable to receive requests of web pages from web browser 23 and to respond by generating and providing the requested web pages. The information content of the web pages may be dynamically obtained by communicating with IA server 14, which is operable to communicate with master controller 36 via control network server 40 to obtain status and other information. Control network server 40 is used only if there is protocol conversion or other control issues needed to operate the control area network. It may be thought of, logically, that IA server 14 is directly coupled to the network and functions as a device on the network. Commands entered at a web browser are sent to web server 13, which relays the commands to master controller 36 via IA server 14 and control network server 40. Master controller 36 then instructs

appropriate appliances and/or systems in the control network to act according to the received command.

The World Wide Web appears to the average user as an enormous data storage of information which is fairly standardized. In reality, web content is often made highly dynamic to the user with various CGI operations. Therefore, most web pages consist of interesting but dynamic content along with a large amount of uninteresting formatting information used to present and display the content.

Referring to FIGURE 3, the present invention of a system and method of mining web page content 60 is operable to receive web pages 62 as input, extract specific content of web pages 62, and output the desired content as processed web pages or files 64 formatted for display on non-PC Internet appliances and devices.

Referring to FIGURE 4, a more detailed block diagram of the system and method of the present invention is shown. System 60 includes a web mining engine 70 which uses templates 72 to determine the location and format of interesting data in web pages 62. Templates 72 are instances of web pages 62, the format of which has been laid out and marked with special tags. The tags in templates 72 guide web mining engine 70 to determine the location of interesting data in new instances of the same web page downloaded from the targeted web site. Web mining engine 70 thus handles the file operations and content extraction. It loads the input and template files 62 and 72 into memory to process the files.

The extracted data are provided to a mined content manager 74, which stores the data in a mined content hierarchical data structure 76, such as a tree data structure, for example. Mined content 76 is stored so that the content can be grouped and easily obtained by a presentation/application layer 78 of system 60. Presentation/application layer 78 can be tailored to specific needs. For example, presentation/application layer 78 may format the data for presentation on a specific Internet appliance, such as the ViewPoint™ touch panel offered by Panja, Inc. Alternatively as another example, presentation/application layer 78 may perform an HTML to XML conversion or filtering function to output the data in XML. Yet another presentation/application layer 78 function may be to place the content directly into a database having a specific format. Presentation/application layer 78 communicates through COM wrappers to mined content manager 74 to retrieve the stored data.

As described above, a template is an instance of a web page that has special tags inserted at key locations to direct the data mining process. According to the present invention, there are three types of tags. The first type is a flow/conditional control tag type, which is used to extract content. A second type of tags is search

tags, which are used to set file pointer positions. A third tag type is extraction tags, which are used to describe what and how located content is to be extracted.

The tags may have a specific predetermined format. For example, the tags may the syntax, {{Tagname 'ATTRIB'=xxx 'ATTRIB'=xxx}}. In this exemplary tag syntax, the brackets, {{ }}, serve as the tag delimiter. Attributes and methods follow the tag name and are separated by one or more spaces. The equal sign, =, is used to set the attribute or method value. Closing tags uses a '/' before the tag name.

In an embodiment of the present invention, an ID attribute is used to allow the assignment of unique names to each tag. The ID value may be the variable name for extracted content. In other cases, the GOTO tag may use the ID attribute to redirect web mining engine programmatic flow to a different position within the page. A LEVEL attribute is used to indicate the hierarchical level within a tree structure. Within a tag block, there can be several sub-tags enclosed. Each of these enclosed sub-tags may contain more sub-tags within themselves. Because of the branching levels that form this parent-child hierarchy, the LEVEL attribute can be used to inform web mining engine 70 as to what level the tag should be viewed in relation to other tags. A PASS attribute informs web mining engine 70 which pass to execute the tag action. When reading an instance page, web mining engine 70 may need to make multiple passes over the page to collect and process information. The PASS attribute uses a numerical value to represent the pass count. For example, if the value is set to 2, then web mining engine 70 would execute the tag action during the second pass. An ONOK attribute is set to the ID value of another tag within the page. When a tag operation is successful, web mining engine 70 is redirected to this tag. In some cases, the output of a successfully tag action may be different. The ONOK attribute allows web mining engine 70 to be redirected depending on the output. An ONFAIL attribute is set to the ID value of another tag within the page. When a tag operation fails, web mining engine 70 is redirected to this tag. In most instances, web mining engine 70 would be directed to an error or status logging action tag.

The following is a discussion of exemplary tags contemplated by the teachings of the present invention.

A NODE tag is used to bound a block of raw text, which contains a set of related content and tags. The NODE tag instructs web mining engine 70 to inform mined content manager 74 that a new node or branch needs to be created. Used in this way, the NODE tag is used to bound content into logical groups within the page. There may be several node blocks denoted on a single web page, and a node block

may be nested within one or more node blocks. The ID attribute for the NODE tag is used to identify the content node stored in mined content tree structure 76.

A LINK tag is used to bound a URL of a sub-page linked to the present page. The sub-page can reside locally with the present page being processed or at another web site. The LINK tag instructs web mining engine 70 to load or initiate another instance of itself. The new instance will load the URL page and the template file passed in the attributes. A child content tree is formed from the linked page content. The parent web mining engine process sleeps until the child engine process has fully mined or scraped the linked page. Once a linked page is processed and the process returns, the child web mining engine instance is shut down and a child content tree is returned. The child content tree is then inserted into the parent tree as a branch node. The linked mining process can continue for multiple levels of linked pages before it returns to the parent engine process. The LINK tag should be used only on links that contain sub-content needing to be extracted. A web page may have one or more links that do not require extraction. When the template is created using the LINK tag, the determination of whether a sub-page should be mined is made. A TEMPLATE attribute of the LINK tag is used to identify the template for the sub-page when parsing the sub-page identified by the URL.

A FIND tag is used to instruct web mining engine 70 to search for a bounded 'string' in the web page. The POS attribute of the FIND tag directs web mining engine 70 where to place the file position pointer once the text is located. The BEGIN attribute of the FIND tag is used to set the file pointer to the first character of the search string. The END attribute sets the file pointer to the first character following the end of the search text. The MOVE attribute instructs web mining engine 70 to move the file pointer to a value +/- n of characters relative to the POS attribute position. If the FIND string is not located, the file pointer remains at the character position where the search began.

A SKIP tag is used to instructs web mining engine 70 to skip over a section of text based on the conditions placed in an EXPRESSION attribute of the SKIP tag.

A GOTO tag is used to instruct web mining engine 70 to move the file pointer to the first character position based on the attribute settings. A MARK attribute of the GOTO tag is used to instruct web mining engine 70 to move the file pointer to a tag of the same ID value. A LINE attribute of the GOTO tag is used to move the file pointer to a specific line number in the web page. The LINE attribute is used when the web page line number is constant.

A MOVE tag is used to instruct web mining engine 70 to move the file position pointer a specific number of characters or lines in either direction. A COUNT attribute sets the number of positions to move. A TYPE attribute instructs web mining engine 70 to move by characters or lines. If the count is outside the physical file boundary, then the file position pointer remains on the last possible character in the direction of the move.

A MARK tag is used to mark a character position in the web page and is often used in conjunction with the GOTO tag. The GOTO tag uses the MARK ID value to instruct web mining engine 70 where to move the file pointer.

A REPEAT tag allows a set of tag commands to be repeated until a conditional statement is met. A COND attribute of the REPEAT tag is used to hold a conditional statement, which may be used to break the loop iteration. A {{BREAK}} tag may also be used within the repeat block to break the loop iteration.

A FOR tag is used to instruct web mining engine 70 to programmatically execute a block of tags 'n' number of times. The FOR tag is typically used to handle repetitive page layouts. If a series of content is identical in format, then a single set of tags can be set up and applied repeatedly using the FOR tag to extract the content. A COUNT attribute may be used to indicate the number of iterations to repeat.

An EXTRACT tag is used to block a section of text to be extracted. A TILL attribute may be used to specify a constant character string that marks the end of the extraction search process. A COUNT attribute may be used to specify the number of characters to extract. RIMREGLEFT and TRIMREGRIGHT attributes are used to specify specific character data to be removed the first and the last characters, respectively, from the extracted string. If no attributes are set, all characters between the EXTRACT and /EXTRACT tags are extracted.

A CALLBACK tag is used to instruct web mining engine 70 to execute a call back function. A ROOT attribute for the CALLBACK tag is used to specify the root node of the content tree to be passed to the call back function.

The use of a PUSHPOS tag allows web mining engine 70 to push the current file pointer onto a stack, and the use of a POPPOS tag is used to retrieve pointers placed on the stack.

An ABORT tag can be used to instruct web mining engine 70 to abort its mining processes immediately. As a result, web mining engine 70 properly closes all currently executing processes before shutting itself down.

A REF tag is used to instruct web mining engine 70 to create a reference to an existing branch of the content tree. Within a control loop or code block, a reference to a portion of the content tree can be created for quick direct access.

Referring to FIGURE 6, an object diagram of the present invention is shown. Web mining engine 70 includes objects 100, where a tag template instance 106 represents a single tag template file. Web mining engine 70 includes a template parser engine 108, which instantiates tag template instance 106 and passes the extracted tag information to a tag class factory 110. Tag class factory 110 creates tag nodes based on the tag data set. After template parser engine 108 extract a tag data from tag template instance 106, tag class factory 110 creates a single node to store the information. The tag node contains the tag values and methods for extracting content from a web page. Each tag contains a specific instruction for either searching or content extraction. Template parser engine 108 builds and orders the tag nodes into a data structure such as a tag node tree 112 based on the type of tags being parsed.

Once the node tree 112 is built, web mining engine 70 loads a web page or scrape file instance 114 for data mining. Page parser engine 116 instantiates scrape file instance object 114 after tag node tree 112 is built. Scrape file instance object 114 represents a single instance of a page to be mined. Tags are read from tag node tree 112 one by one and the tag action is performed on the current instance of the web page to locate and extract content. It maintains a set of internal file pointers and a special pointer stack used by page parser engine 116. Any extracted content is immediately passed to mined content tree manager 102, which stores the mined data in a mined content tree 118, which is a hierarchical arrangement of the extracted web content data. Because both tag node tree 112 and mined content tree 118 are built based on the tags, their structure is the same or similar to one another.

An example 80 of mined content tree 118 is shown in FIGURE 5. It may be seen that this structure allows grouping of related content nodes. Exemplary tree 80 has named nodes which are indicative of the type of content that is stored at each node. For example, a root node 81 is directly connected to a news node 82, a stock node 83, and a music node 84. News node 85 is in turn connected to sports node 85 for sports news and to weather node 89 for weather forecasts. Sports node 85 may be further connected to football, baseball, and hockey nodes 86-88, and weather node 89 may be further connected to Dallas weather node 90 and Texas weather node 91, for example. Music node 84 may be connected to an MP3 node 92 to receive streaming music. To traverse the tree structure, a current node pointer is used. When content is requested by the presentation/application layer, the content of the current node pointer

is delivered. If content from another node is requested, the current node pointer can be set to the new node or a relative path string to the requested node can be used without changing the current node pointer position.

Once the mining operation is complete, presentation/application layer 104 uses its objects 120 to interface with the mined content manager. Presentation/application layer objects 120 may generate output such as XML pages, storing content into a database, or arranged in a format suitable for display on a particular Internet appliance, for example.

The present invention is a targeted web mining system and method because the layout of a web page must first be analyzed and tagged to create a template before its content can be extracted. Because some display formats used by non-PC devices, such as Internet appliances, touch screen tablets, personal digital assistants cannot display information layout designed for a PC monitor, the present invention allows such non-PC devices to extract and display targeted web content. The presentation/application layer may reformat the extracted data to fit the screen layout of such devices. Alternatively, the presentation/application layer may perform a translation function and translate the mined web content into another markup language and/or another international language.

The presentation/application layer may alternatively use the extracted web content to perform a specific function. For example, if the extracted web content is the local TV programming schedule, the presentation/application layer may directly use the extracted information to program a television set or video cassette recorder. As another example, if the extracted web content is the local weather forecast, the presentation/application layer may directly set the lawn sprinkler system to rain if rain is forecasted.

Although several embodiments of the present invention and their advantages have been described in detail, it should be understood that mutations, changes, substitutions, transformations, modifications, variations, and alterations can be made therein without departing from the teachings of the present invention, the spirit and scope of the invention being set forth by the appended claims.

WHAT IS CLAIMED IS:

1. A method of extracting content from a web page, comprising:

loading a template of the web page and parsing at least one set of tags in the template;

parsing an instance of the web page and extracting content according to the parsed tags;

storing the extracted content into a hierarchical data structure; and

processing the extracted content for output.

2. The method, as set forth in claim 1, wherein creating the template comprises inserting at least one set of tags for flow control.

3. The method, as set forth in claim 1, wherein creating the template comprises inserting at least one set of tags for setting a file pointer used during web page parsing.

4. The method, as set forth in claim 1, wherein creating the template comprises inserting at least one set of tags for indicating how demarcated web page content is to be extracted.

5. The method, as set forth in claim 1, wherein creating the template comprises inserting at least one set of node tags to logically group a block of data on the web page and to store the block of data into a logical unit of the second hierarchical data structure.

6. The method, as set forth in claim 1, wherein creating the template comprises inserting at least one set of link tags to demarcate a URL link to a second web page to be processed.

7. The method, as set forth in claim 1, wherein creating the template comprises inserting at least one set of find tags to specify a data string to search for in the web page.

8. The method, as set forth in claim 1, wherein creating the template comprises inserting at least one set of skip tags to specify a block of data to ignore.

9. The method, as set forth in claim 1, wherein creating the template comprises inserting at least one set of goto tags to move a file pointer to a specific point in the web page.

10. The method, as set forth in claim 1, wherein creating the template comprises inserting at least one set of extract tags to specify a block of data to be extracted.

11. The method, as set forth in claim 1, wherein creating the template comprises inserting at least one set of tags to control iterative execution control.

12. The method, as set forth in claim 1, further comprising storing the at least one set of tags in a first hierarchical data structure such as a tree data structure.

13. The method, as set forth in claim 1, wherein storing the extracted content into a second hierarchical data structure comprises storing the extracted content in a tree data structure.

14. The method, as set forth in claim 1, wherein processing the extracted content for output comprises:

reading the stored content from the second hierarchical data structure; and

reformatting the extracted content for display on a specific device.

15. The method, as set forth in claim 1, wherein processing the extracted content comprises translating the extracted content in one language to a second language.

16. The method, as set forth in claim 1, further comprising creating the template of the web page, the template having the at least one set of tags to demarcate web content and control content extraction.

17. A method of extracting content from a targeted web page, comprising:

creating a template of the targeted web page, the template having the at least one set of tags to demarcate web content to be extracted and control content extraction, the at least one set of tags includes tags for flow control, tags for setting a file pointer used during web page parsing, and tags for indicating how demarcated web page content is to be extracted;

parsing the at least one set of tags in the template;

storing the parsed at least one set of tags in a first data structure;

parsing an instance of the web page and extracting content according to the stored tags in the first data structure;

storing the extracted content into a second data structure; and

accessing the extracted content stored in the second data structure and processing the extracted content for output.

18. The method, as set forth in claim 17, wherein creating the template comprises inserting at least one set of node tags to logically group a block of data on the web page and to store the block of data into a logical unit of the second hierarchical data structure.

19. The method, as set forth in claim 17, wherein creating the template comprises inserting at least one set of link tags to demarcate a URL link to a second web page to be processed.

20. The method, as set forth in claim 17, wherein creating the template comprises inserting at least one set of find tags to specify a data string to search for in the web page.

21. The method, as set forth in claim 17, wherein creating the template comprises inserting at least one set of skip tags to specify a block of data to ignore.

22. The method, as set forth in claim 17, wherein creating the template comprises inserting at least one set of goto tags to move a file pointer to a specific point in the web page.

23. The method, as set forth in claim 17, wherein creating the template comprises inserting at least one set of extract tags to specify a block of data to be extracted.

24. The method, as set forth in claim 17, wherein creating the template comprises inserting at least one set of tags to control iterative execution control.

25. The method, as set forth in claim 17, wherein storing the at least one set of tags in a first hierarchical data structure comprises storing the tags in a tree data structure.

26. The method, as set forth in claim 17, wherein storing the extracted content into a second hierarchical data structure comprises storing the extracted content in a tree data structure.

27. The method, as set forth in claim 17, wherein processing the extracted content for output comprises:

reading the stored content from the second hierarchical data structure; and

reformatting the extracted content for display on a specific device.

28. The method, as set forth in claim 17, wherein processing the extracted content comprises translating the extracted content in one language to a second language.

29. A system of mining targeted world wide web content, comprising:

a template created for each targeted web page, each template including at least one set of tags for flow control and content extraction;

a template parsing engine operable to parse the at least one set of tags in the template and storing the tags in a first data structure;

a page parsing engine operable to parse an instance of the targeted web page corresponding to the template according to the stored tags in the first data structure, and extracting its content;

a second data structure used to store the extracted content of the targeted web page; and

a presentation application operable to access, process and output the stored content.

30. The system, as set forth in claim 29, wherein the first data structure is a tree having at least one node, each node storing a set of tags and associated attribute data.

31. The system, as set forth in claim 29, wherein the second data structure is a tree having at least one node, each node storing a logical grouping of extracted web content.

32. The system, as set forth in claim 29, wherein the template comprises at least one set of tags for setting a file pointer used during web page parsing.

33. The system, as set forth in claim 29, wherein the template comprises at least one set of tags for indicating how demarcated web page content is to be extracted.

34. The system, as set forth in claim 29, wherein the template comprises at least one set of node tags to logically group a block of data on the web page and to store the block of data into a logical unit of the second hierarchical data structure.

35. The system, as set forth in claim 29, wherein the template comprises at least one set of link tags to demarcate a URL link to a second web page to be processed.

36. The system, as set forth in claim 29, wherein the template comprises at least one set of find tags to specify a data string to search for in the web page.

37. The system, as set forth in claim 29, wherein the template comprises at least one set of skip tags to specify a block of data to ignore.

38. The system, as set forth in claim 29, wherein the template comprises at least one set of goto tags to move a file pointer to a specific point in the web page.

39. The system, as set forth in claim 29, wherein the template comprises at least one set of extract tags to specify a block of data to be extracted.

40. The system, as set forth in claim 29, wherein the template comprises at least one set of tags to control iterative execution control.

41. The system, as set forth in claim 29, further comprising a display device operable to display the extracted and processed web content.

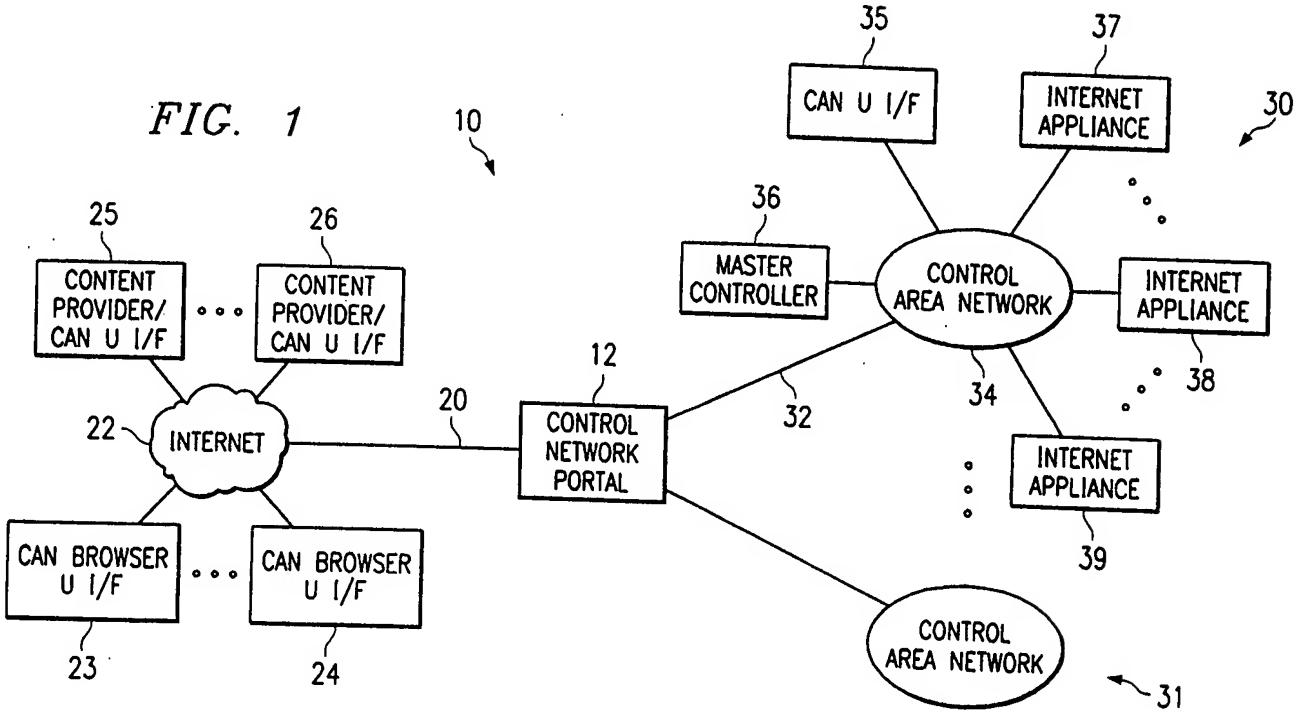
42. The system, as set forth in claim 29, further comprising an Internet appliance operable to display the extracted and processed web content.

SYSTEM AND METHOD OF MINING WORLD WIDE WEB CONTENT

ABSTRACT OF THE INVENTION

A system and method of mining targeted world wide web content are provided. The system includes a template created for each targeted web page, where each template including at least one set of tags for flow control and content extraction. A template parsing engine is operable to parse the at least one set of tags in the template and storing the tags in a first data structure. A page parsing engine is operable to parse an instance of the targeted web page corresponding to the template according to the stored tags in the first data structure, and extracting its content. A second data structure is used to store the extracted content of the targeted web page. A presentation application is operable to access, process and output the stored content.

FIG. 1


10
12
20
22
INTERNET
CONTROL NETWORK PORTAL
25
CONTENT PROVIDER/ CAN U I/F
26
CONTENT PROVIDER/ CAN U I/F
23
CAN BROWSER U I/F
24
CAN BROWSER U I/F
30
35
CAN U I/F
37
INTERNET APPLIANCE
36
MASTER CONTROLLER
CONTROL AREA NETWORK
INTERNET APPLIANCE
38
32
34
INTERNET APPLIANCE
39
CONTROL AREA NETWORK
31

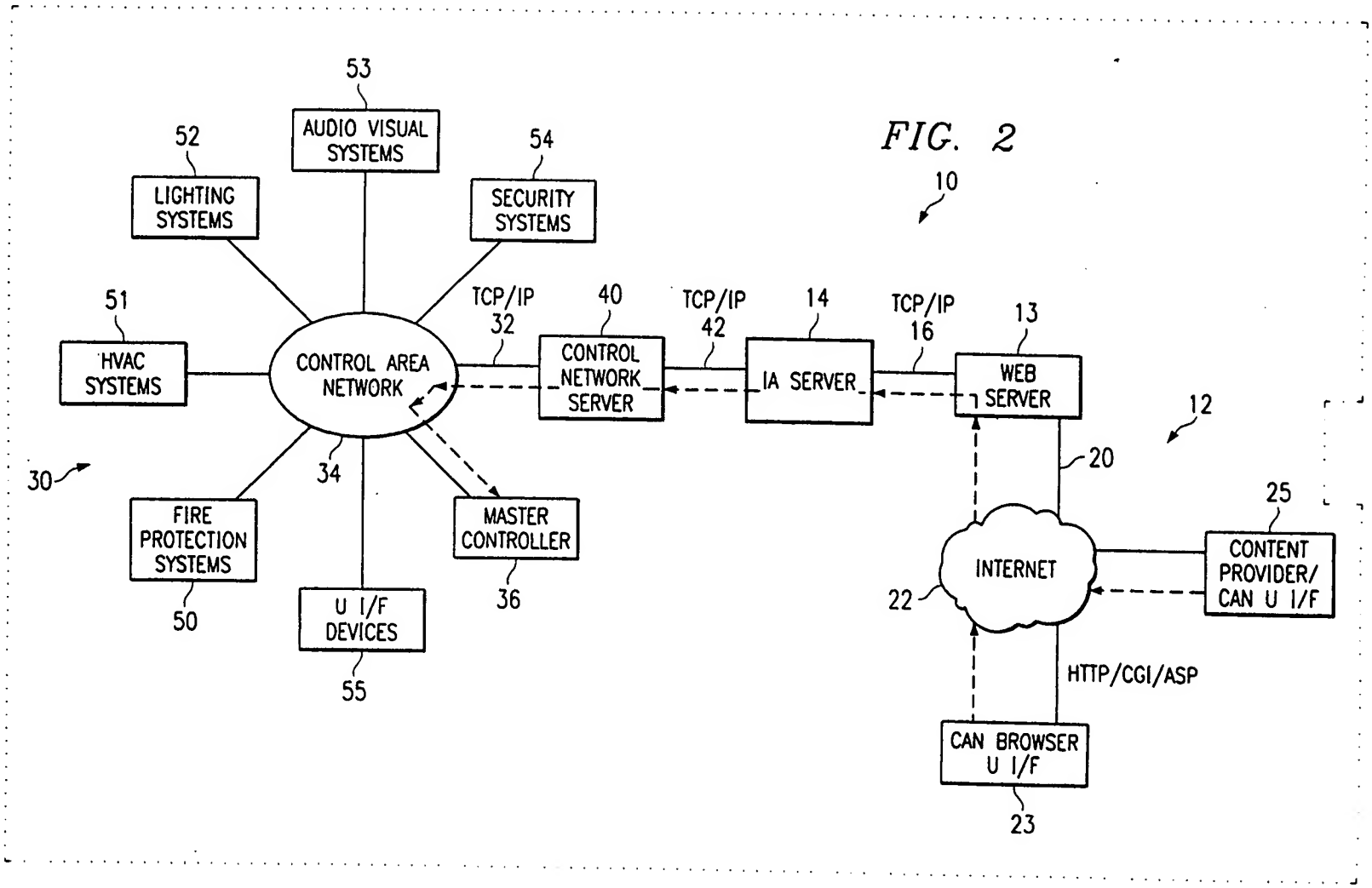

53
AUDIO VISUAL SYSTEMS
52
LIGHTING SYSTEMS
54
SECURITY SYSTEMS
51
HVAC SYSTEMS
CONTROL AREA NETWORK
TCP/IP
32
40
CONTROL NETWORK SERVER
TCP/IP
42
14
IA SERVER
TCP/IP
16
13
WEB SERVER
10
12
30
34
20
25
FIRE PROTECTION SYSTEMS
MASTER CONTROLLER
INTERNET
CONTENT PROVIDER/ CAN U I/F
22
50
U I/F DEVICES
36
HTTP/CGI/ASP
CAN BROWSER U I/F
55
23

FIG. 2

FIG. 3

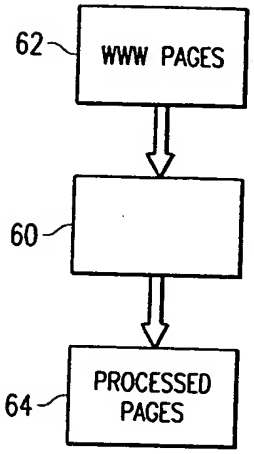


FIG. 4

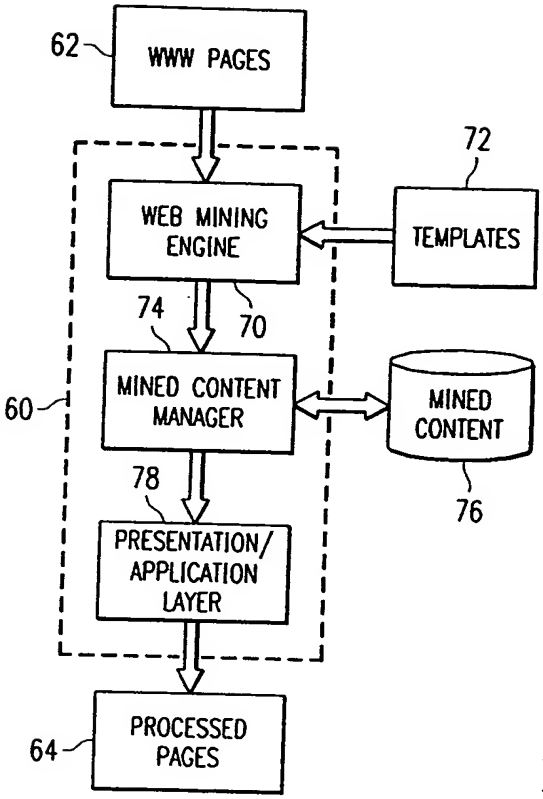


FIG. 5

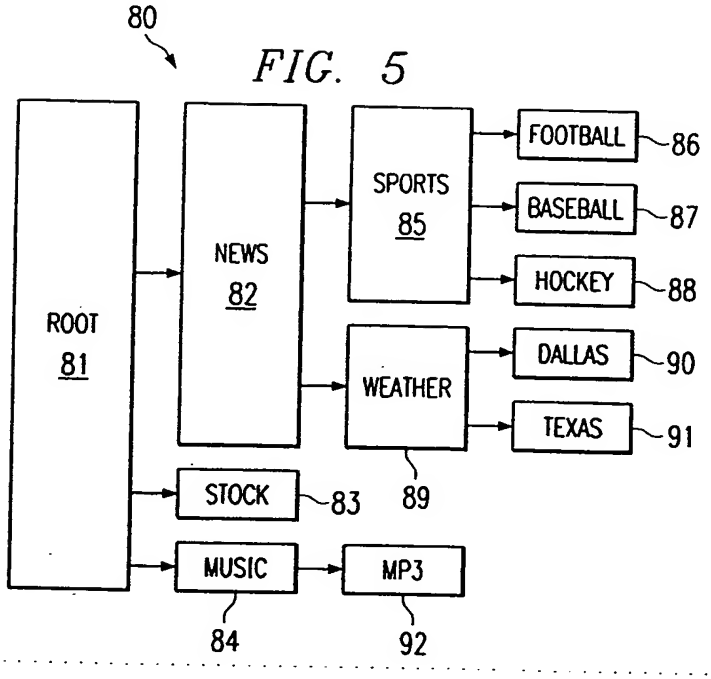
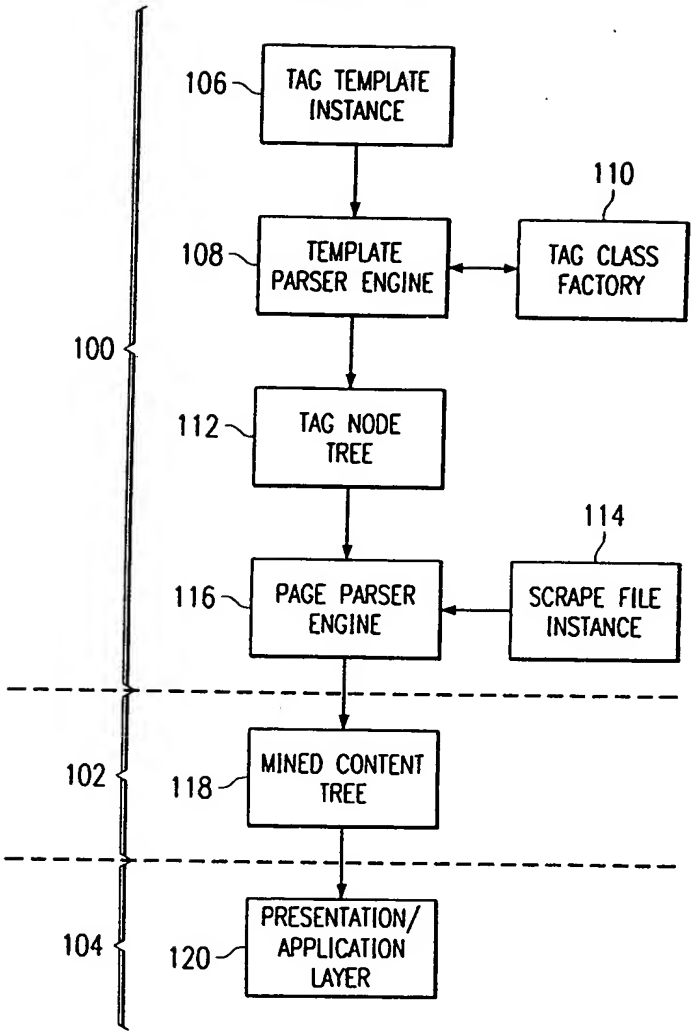


FIG. 6


106
TAG TEMPLATE INSTANCE
110
108
TEMPLATE PARSER ENGINE
TAG CLASS FACTORY
100
112
TAG NODE TREE
114
116
PAGE PARSER ENGINE
SCRAPE FILE INSTANCE
102
118
MINED CONTENT TREE
104
120
PRESENTATION/ APPLICATION LAYER